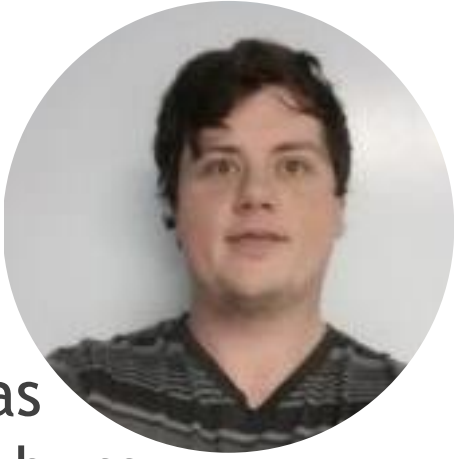


Nicholas
Blackburn



2018

Smartsettia

Greenhouse Automation / App Controlled
Poinsettia Covering System

Dakota
Jackson



Maxwell
Thornburg



What is Smartsettia?

- ▶ The University of Idaho Greenhouse grows poinsettias
- ▶ Poinsettias have an optimal time for light exposure
- ▶ In 2016, a team used a garage door motor to cover poinsettias
- ▶ In 2017, the first Smartsettia team created a basic front-end
- ▶ This year we focused on:
 - Management (Access Controls)
 - Visuals (Streaming and Real Data)
 - Security (Motion Detection)



Outline



Problem Statement



Streaming Live Video



Sensor Communication



Motion Detection



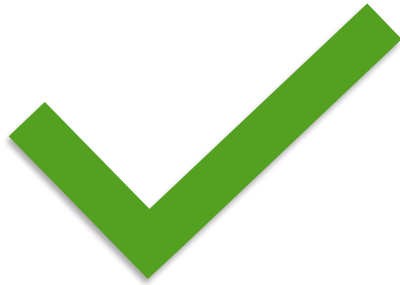
Device Level Access



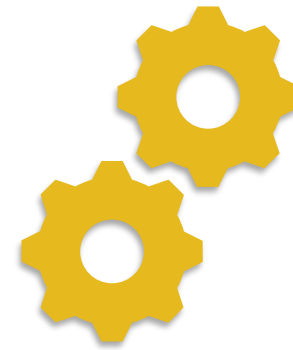
Backend rework



Problem Statement



Make improvements and add functionality to the existing Smartsettia environment.



Transform the specific greenhouse application into general automated system.



Streaming Live Video

with RTMP to DASH / HLS

- ▶ Why not stream to a port and listen?
- ▶ WebRTC or RTMP?
- ▶ RTMP, use FFMPEG to stream
- ▶ NGINX Reinstallation with RTMP Package
- ▶ Redirecting RTMP to DASH / HLS
- ▶ Default Video.js Player Framework
- ▶ Request Stream

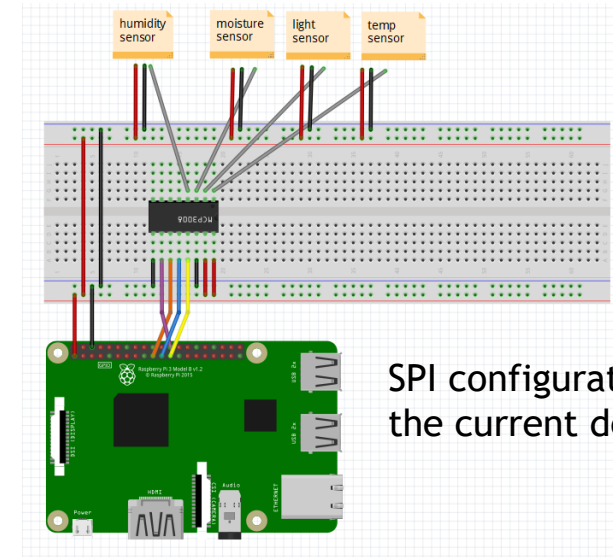
```
rtmp {  
    server {  
        listen 1935;  
        chunk_size 8192;  
  
        application live {  
            play /usr/local/nginx/rtmp;  
        }  
    }  
}
```

Browser	DASH	HLS
Firefox 32	✓ [1]	✓ [2]
Safari 6+		✓
Chrome 24+	✓ [1]	✓
Opera 20+	✓ [1]	
Internet Explorer 10+	✓ 11	✓ [2]
Firefox Mobile	✓	✓
Safari iOS6+		✓
Chrome Mobile	✓	✓ [2]
Opera Mobile	✓ [1]	✓
Internet Explorer Mobile	✓ 11	✓ [2]
Android	✓	

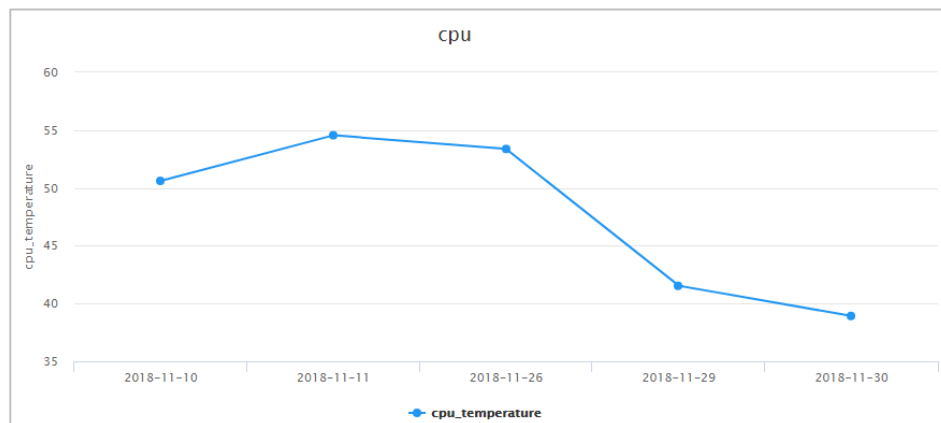


Sensor Communication over I2C, SPI, GPIO

- ▶ Communication between the pi & sensors:
 - ▶ I2C, SPI (Serial Peripheral Interface), or GPIO
- ▶ Smartsetti API
- ▶ Sensor Controller
- ▶ Displaying the sensor data



SPI configuration of
the current device



Test Device

Name:	Type:	Value:	Updated At:
cpu	cpu_temperature	39.7	53 seconds ago
light_in	light	959	53 seconds ago
light_out	light	959	53 seconds ago
temperature	temperature	-58.0	53 seconds ago
humidity	humidity	1012	53 seconds ago
moisture	moisture	1023	53 seconds ago



Motion Detection with OpenCV

- ▶ Motion detection using a camera (not hardware)
- ▶ OpenCV is a Computer Vision Library
- ▶ How do we use OpenCV to detect motion?
- ▶ Trivial task
 - ▶ Greyscale
 - ▶ Blur
 - ▶ Save Image
 - ▶ Repeat
 - ▶ Compare Images
 - ▶ Find Contours

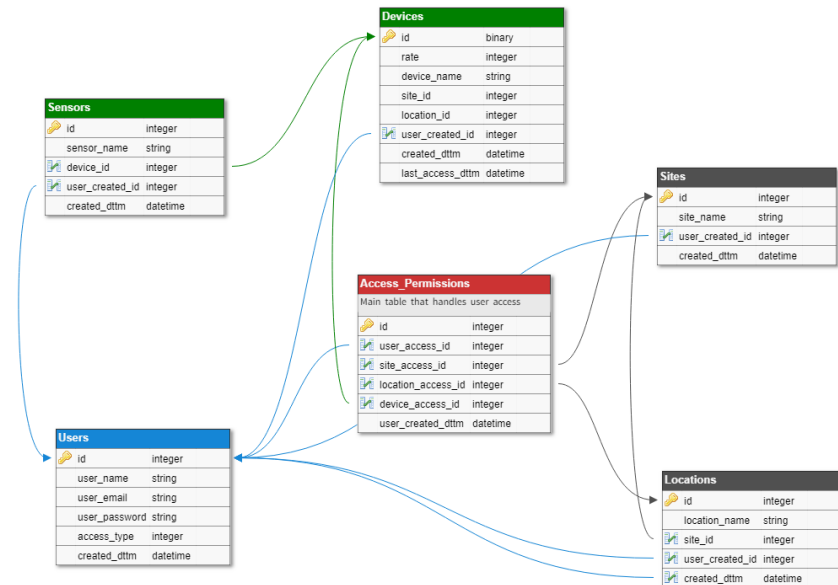


A theoretical example of an image OpenCV would end up with to do motion detection.



Device Level Access

- ▶ Created a new database table Access_Permissions
- ▶ Created a User.php with functions for:
 - ▶ Deleting and inserting rows into Access_Permissions
 - ▶ Selection queries for which devices are available to the current user
 - ▶ Interacts with the existing User structure
 - ▶ The helper functions to accompany the previous
- ▶ Ideas for improvement:
 - ▶ More details such as who gave permissions
 - ▶ Granular access: edit, rename, configure sensors



Backend Rework

Laravel Best Practices

- ▶ Reduced the number of blades
- ▶ Created a Master Blade
- ▶ Implemented functions into accompanying blades
- ▶ Updated Data Tables



Thanks.
Questions?

